

## **IMAGE PROCESSING UTILIZING LOCAL COLOR CORRECTION AND CUMULATIVE HISTOGRAMS**

### **BACKGROUND**

**[0001]** Conventional local color correction algorithms include nonlinear masking, Retinex and its derivatives, and piecewise gamma estimation. These algorithms are useful for improving poorly rendered digital images, images with large dynamic ranges, and images with poor illumination. For example, an image captured by a digital camera is underexposed and dark. The local color correction can be used to correct the image for the underexposure.

**[0002]** These conventional algorithms are not convergent in all cases. That is, these conventional algorithms do not provide a unique solution: they do not indicate the amount of color correction (if any) that provides the best image quality. Instead, they provide multiple possible solutions, all of which depend upon input parameters. Moreover, repeated application of a particular algorithm can degrade the overall image quality.

**[0003]** There is a need for a parameter estimate that can determine whether an image should be color-corrected and, if so, by what amount. Ideally, the parameter estimate should be small or zero if a local color correction algorithm has already been applied. Likewise, the parameter estimate should be small or zero if the image does not require local color correction. The parameter-estimate should also be able to detect corner cases of poorly rendered data and images in obvious need of local color correction, and to avoid correcting those images that would not benefit from local color correction.

**SUMMARY**

**[0004]** According to one aspect of the present invention, processing of an image includes estimating a parameter for local color correction. The parameter estimation includes computing a cumulative histogram path length of the image. Other aspects and advantages of the present invention will become apparent from the following detailed description, taken in conjunction with the accompanying drawings, illustrating by way of example the principles of the present invention.

**BRIEF DESCRIPTION OF THE DRAWINGS**

**[0005]** Figure 1 is an illustration of a method of performing parameter estimation in accordance with an embodiment of the present invention.

**[0006]** Figures 2-6 are illustrations of a simple example of computing a cumulative histogram path length.

**[0007]** Figure 7 is an illustration of a method of performing image processing in accordance with a first embodiment of the present invention.

**[0008]** Figure 8 is an illustration of a method of performing image processing in accordance with a second embodiment of the present invention.

**[0009]** Figure 9 is an illustration of a method of performing image processing in accordance with a third embodiment of the present invention.

**[0010]** Figure 10 is an illustration of a system in accordance with an embodiment of the present invention.

**DETAILED DESCRIPTION**

**[0011]** The present invention is embodied in a method for processing a digital image. A parameter estimate is generated using a cumulative histogram path length of an image. The parameter estimate can be used to determine whether the image should be color-corrected and, if so, the amount of color correction. The parameter estimate is small or zero if a local color correction algorithm has already been applied so as to avoid repeated application of the local color correction algorithm.

**[0012]** Reference is made to Figure 1, which illustrates a method of performing parameter estimation for a digital image. A raw histogram of the image is computed (110). The raw histogram may be computed by splitting the pixel range into equal-sized bins and, counting the number of pixel values that fall into each bin. In general, the raw histogram is shown by color component and the number of bins is a function of the bit depth. For example, the histogram for an 8-bit image has 256 bins, the histogram for a 16 bit image (grayscale, RGB, YUV) has 65535 bins and so on.

**[0013]** A cumulative histogram is generated from the raw histogram (112). The  $n^{\text{th}}$  bin in the cumulative histogram may be computed by adding the count of the  $n^{\text{th}}$  bin in the raw histogram to the count of the  $n-1^{\text{th}}$  bin in the cumulative histogram. Thus a bin in the cumulative histogram represents the number of pixels in the image with a value less than or equal to the bin value.

**[0014]** The cumulative histogram is normalized (114). The normalized cumulative histogram may be computed by dividing cumulative histogram by the number of pixels in the image and digital counts by the maximum pixel value.

**[0015]** A cumulative histogram path length (CHPL) is computed from the normalized cumulative histogram (116). The CHPL is the cumulative sum of the distances between neighboring points in the normalized cumulative histogram.

**[0016]** The CHPL can be used to estimate whether color correction should be performed and, if so, how much local color correction should be applied. The value roughly corresponds to how bi-modal the raw histogram is: the lower the value, the closer to a uniform histogram; and the higher the value, the closer to a

bi-modal histogram. For example, for a cumulative histogram normalized to the range 0 to 1, the values of the CHPL range from the square root of two or roughly 1.414, for an image with a flat histogram, to 2.0, for an image consisting of a single digital count. These examples are extremes; in general, natural scenes can have values somewhere between these two extremes.

**[0017]** Consider the following example of computing a CHPL for the 3x2 pixel image shown in Figure 2. The image has 8-bit pixel values. The corresponding raw histogram is shown in Figure 3. The histogram has five bins, with two pixels falling in the first bin, and one pixel falling in each of the second, third, fourth and fifth bins. The cumulative histogram is illustrated in Figure 4, and the normalized cumulative histogram is illustrated in Figure 5. Note that this is an extremely simplified example. In general, images are considerably larger and can have a significant number of different pixel values.

**[0018]** As shown in Figure 6, the CHPL is computed as a distance between each of the points in the cumulative histogram path length. Specific to Figure 6, the CHPL is computed as the lengths of the horizontal and vertical line segments. The lengths of the line segments shown in Figure 6 add up to  $(0.1255+0.33+0.1176+0.1667+0.1454+0.1667+0.2117+0.1667+0.1765+0.1667+0.2039)=1.976$ .

**[0019]** The method is not limited to computing the CHPL as the sum of sides, as shown in Figure 6. The CHPL could be computed in other ways. For example, the CHPL could be computed as a sum of hypotenuses; or it could be computed by fitting a curve to the cumulative histogram, and determining the curve length.

**[0020]** This method can be applied to a variety of color encodings. It can work with grayscale, RGB, YUV and most commonly used color spaces. If the digital image is represented in YUV space, the cumulative histogram can be computed from the Y color component, and the corresponding strength is used with the local color correction as applied to the Y color component. If the digital image is a grayscale image, the strength of the local color correction applies to the single grayscale component.

**[0021]** If the digital image is represented in a non-perceptual space such as RGB, a cumulative histogram is determined for each color component, and an average cumulative histogram is determined by averaging the cumulative histograms for the three color components. This average cumulative histogram is then used to determine a strength for the local color correction, and local color correction is performed on each of the color components independently.

**[0022]** Reference is now made to Figure 7, which illustrates a first example of using the CHPL. The CHPL is computed for a digital image (710), and used to determine whether color correction should be performed (712). The decision to perform color correction can be based on assumptions about the value of the CHPL. For example, an assumption may be made that if the value of a CHPL is greater than an upper threshold (e.g., 1.65), then that CHPL is not a natural scene (e.g., it is computer generated) and does not require correction. Similarly, another assumption may be made that if the value of a CHPL is below a lower threshold (e.g., 1.5), then that CHPL is well rendered and does not require correction. Thus, the color correction would be performed if the CHPL is within a range defined by upper and lower thresholds.

**[0023]** If the cumulative histogram path length indicates that the image should be color corrected (714), the color correction is performed (716). The color correction may be performed by using a conventional local color correction algorithm such as nonlinear masking, Retinex, or piecewise gamma estimation may be used. Other algorithms that may be used are disclosed in U.S. Serial No. 10/136,966 filed May 2, 2003 entitled "Local Color Correction" and incorporated herein by reference.

**[0024]** Reference is now made to Figure 8, which illustrates a second example of using the CHPL. An image 810 is color corrected at different "strengths" (812). The strength is the amount or aggressiveness of the local color correction. The color correction at this stage can be an approximation of the local color correction that will be performed later. This approximation can be performed by applying the local color correction algorithm to the image at different strengths.

**[0025]** The approximation is not limited to any particular local color correction algorithm. However, the sigmoidal-exponential tone reproduction disclosed in U.S. Serial No. U.S. Serial No. 10/136,966 can be implemented efficiently as a series of different lookup tables ( $LUT_1$ ,  $LUT_2$ , ...) for different strengths (e.g., 0, 64, 128, 192, and 255).

**[0026]** A plurality of CHPLs are computed, one for each color-corrected image (block 814). A CHPL is also computed for an image that is not color corrected. The CHPL having the lowest value (that is, the minimum histogram path length) is selected (block 816). Many images will have the same cumulative histogram path length, but not all images with that cumulative histogram path length will need local color correction. For an image not needing color correction, the LUTs will produce a set of cumulative histogram path lengths that are longer than the CHPL for the uncorrected image. Since the CHPL corresponding to no color correction will be shortest, these image will not be color corrected. However for an image needing color correction, one of the outputs from the LUTs will have the shortest cumulative histogram path length. The correction strength corresponding to this minimum is used to perform local color correction of the image, if any (818). The local color correction at block 818 is not an approximation.

**[0027]** An upper bound on the minimum CHPL can be determined (820). It is possible in some cases, such as driver processing, that a highly bimodal image (e.g., a bar chart) would have a very high cumulative histogram path length, but would not likely benefit from local color correction. The upper bound can minimize the chance of processing graphics or images that are highly bi-modal by nature, such as astronomical images with a majority of black or low pixel values.

**[0028]** Figure 9 illustrates a third example of using the CHPL. The third example is essentially the same as the second example, except as follows. First, the approximation is inferred from a thumbnail 911 of the original image 910. The use of the thumbnail 911 allows an even faster estimation of the cumulative histogram path length. For processing pipelines that include thumbnails, such as digital cameras, the use of thumbnails instead of full sized images can be considerably faster.

**[0029]** Second, the approximation is performed by using multiple settings of different local color correction algorithms 912. For example, the approximation may be performed by piece-wise gamma at different strengths, Retinex at different strengths, and different strengths of the local color correction algorithm disclosed U.S. Serial No. 10/136,966. In the alternative, the approximation may be performed by applying the same algorithm at different strengths.

**[0030]** Afterward a minimum CHPL is computed for the thumbnail (914-916), an upper bound is placed on the minimum CHPL (920), and local color correction, if any, is performed on the original image (918).

**[0031]** The method according to the present invention can be useful for enhancing or improving several types of images. For example, conventional photography may yield an image in which one region of the image is underexposed while another region is correctly exposed. Flash photography is another instance in which one region in the image may be over exposed while another is under exposed. For these situations, a complex local correction technique based on the human visual system may be inappropriate. In addition, some image scenes have a wide dynamic range. The method provides a solution to improve the image quality in these situations.

**[0032]** The method according to the present invention can be implemented at several stages of an image-processing pipeline. Consider the system 1010 shown in Figure 10, which includes an imaging device 1012 (*e.g.*, a digital camera, a scanner) and a computer 1014. The method according to the present invention may be performed in the imaging device 1012 or the computer 1014. The imaging device 1012 or the computer 1014 can correct the color tone of images that were captured with uneven exposure or flash illumination, post-process images with a high dynamic range, etc.

**[0033]** A program 1016 can be stored in memory 1018 of the imaging device 1012. When executed, the program 1016 can instruct a processor 1020 of the imaging device 1012 to perform the method according to the present invention.

**[0034]** A set of instructions can be stored in memory 1022 of the computer 1014. When executed, the instructions can instruct a processor 1024 of the computer 1014 to perform the method according to the present invention. The instructions can be part of an image processing application 1026 or it could be part of another application (such as a web browser application). The instructions could even be implemented as part of a device driver 1028 such as a printer driver, so that the driver performs correction on an image before a display device 1030 displays the color-corrected image.

**[0035]** The method according to the present invention can be used to complement traditional pixel-based processing. For example, once the image is color corrected, the color-corrected image can be reproduced on a printer using a traditional pixel-based pipeline.

**[0036]** Although several specific embodiments of the present invention have been described and illustrated, the present invention is not limited to the specific forms or arrangements of parts so described and illustrated. Instead, the present invention is construed according to the claims the follow.